Real-Time Dynamic Cloud Monitoring System Based On SLA

Li Yunshui*1, Deng Yule2, Gao junxiao2

Hangzhou New Century Information Technology Company*1, Hangzhou, China

Zhejiang University of Technology2, Hangzhou, China

*1liys@nci.com.cn; 2rainerdun@live.cn; 3gjx8010@hotmail.com

Abstract

In order to solve the problem that there is no suitable SLA system for cloud environment, a real-time dynamic cloud monitoring system has been proposed based on SLA, consisting of SLA agreement information database, distributed cluster data monitoring system, distributed runtime monitoring system, QoS knowledge base and resource scheduler. A prototype of the proposed system has been developed as well as the test has been conducted. The experimental results show that this system, while monitoring the cloud, can reduce the risk of a SLA agreement violation threats by taking actions ahead. In addition, it has a relatively high reliability and scalability.

Keywords

SLA, Cloud Computing; Distributed Cluster Data Monitoring; Distributed Runtime Monitoring

Introduction

Cloud computing allows users to employ the resources in the cloud environment depending on their demand, and only the used is charged. This method greatly reduces the cost of the initiation and maintenance of IT infrastructure as well as the threshold for SMEs to use IT equipment. With a service level agreement (SLA) signed with customers, Cloud service providers now can offer the non-local network resources in a cheaper, more targeted, and rapid way for the users and ensure the interests of both sides. A flexible SLA cloud infrastructure can not only be used to provide guaranties for the user but also facilitate cloud service providers to manage cloud infrastructure resources more effectively. However, the large scale of the cloud computing center, the possible heterogeneity between servers and the uneven load between nodes are all serious threats to the machines' performance and therefore, the quality of the service can't be guaranteed. Therefore, the cloud service providers need to pay a big penalty for the breach of contract.

For massive cloud resource data and SLA protocol data, the existing monitoring system can't satisfy the requirement of real-time and dynamic monitoring, and service default phenomenon often happens. In order to improve the quality of cloud services, and protect the interests of the cloud service providers and users, this paper proposes a real-time dynamic cloud monitoring system based on SLA. The system uses Chukwa as the resource gathering platform, and stores the data in HBase. Meanwhile, the system analyzes and monitors the implementation of SLA, once confronting a violation threats, the system dynamically rearranges the resources on the platform with the help of a resource scheduler based on a QoS knowledge base. Experiments show that the system can real-time collect resource metrics data, dynamically analyzes the implementation of SLA, and effectively avoid some SLA violation threats, guarantee the efficiency of cloud services. The system at the same time used on thousands of nodes in the data collection and analysis, provides pluggable components, and therefore it is easy to customize and enhance functionality.

This paper is organized as follows: the present research status of SLA on a cloud environment is discussed in Section 2. Then the real-time dynamic cloud monitoring system is presented based on SLA in Section 3. Section 4 discusses the performance of our system. Summary and future discuss are in Section 5.

Related Works

SLA based on cloud computing service confronts multiple challenges, such as the algorithm of scalability definition and optimized control, scalable monitoring and the accuracy of monitoring on a distributed system, cloud computing resource reconfiguration, etc. Most of the existing monitoring systems are not directly compatible with the cloud computing platform due to their heavy dependence on the internet or the service oriented infrastructure monitoring.

Comuzzi and his companion proposed an SLA within the framework of the EU project SLA@SOI by means of historical data to assess the SLA having the ability to monitor SLA entries. But they neglected mapping low-level monitoring indicators to high-level SLA parameters, which can be used to monitor the SLA object. NetLogger is a distributed monitoring system, but it is just monitoring network resources. Theilman discussed the multi-level SLA management and the development and management of SLA within the service-oriented infrastructure. They put forward the concept of using the runtime function view architecture to manage the SLA. Frutos and his companion developed a framework on the basis of the EU project BREIN which inherits the characteristics of grid computing and can perform advanced SLA management. However, it can only be used in grid computing. Because most of the owners of resources in the grid are individuals or enterprises, resource availability varies greatly. However, in cloud computing, the owner of the resources are loud service providers, and users' resources are ranged by what is purchased, and resources supply is more stable. Therefore, SLA violation threats of detection method for the grid cannot be directly applied to the cloud environment.

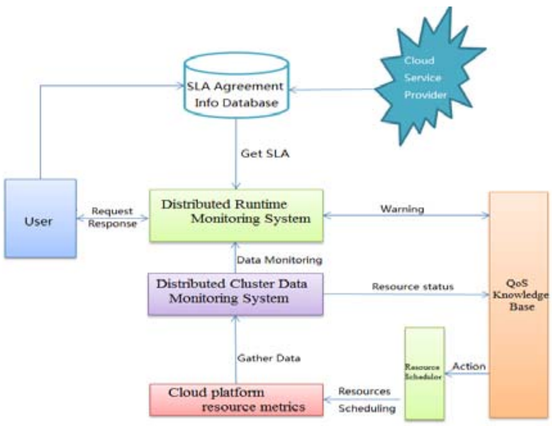

FIG 1 STRUCTURE OF REAL-TIME DYNAMIC CLOUD MONITORING SYSTEM BASED ON SLA

Real-time Dynamic Cloud Monitoring System Based on SLA

In a cloud computing environment, the use of services and resources is elastic from the perspective of the users, so the cloud service provider need to monitor the cloud resource metrics in real-time. By analyzing the advantages and disadvantages of the existing monitoring system, this paper presents a real-time dynamic cloud based on SLA monitoring and control system. System structure diagram is as shown in Fig 1.

Real-time dynamic cloud monitoring system based on SLA consists of a SLA agreement information database, a distributed cluster data monitoring system, a distributed runtime monitoring system, a QoS knowledge base and a resource scheduler.

SLA Agreement Information Database

The database contains the SLA agreement between the service requester and the service responder. Once the customer required providing the agreed service, service providers based platform as a service to the resources offer users with services, including cloud hosting service and network resources, etc.

Table 1 lists some of the SLA parameters on Hadoop, such as availability, a parameter measured according to downtime and uptime.

TABLE 1 SLA PARAMETERS SAMPLE

SLA parameters	Range
Incoming bandwidth (IB)	>10 Mbit/s
Outgoing bandwidth (OB)	>12 Mbit/s
Storage (St)	>1024 GB
Availability (Av)	≥99%
Response Time	Rtotal=Rin+Rout (ms)

Distributed Cluster Data Monitoring System

High scalability and availability are the two main purposes of the system. In the main time, monitoring and analysis on system must support management decisions. Chukwa is a large data acquisition and analysis system based on Hadoop cloud platform developed by Yahoo, and the processing ability on heterogeneous data, the adapter pattern framework mechanism and the hierarchical processing method are of great use.

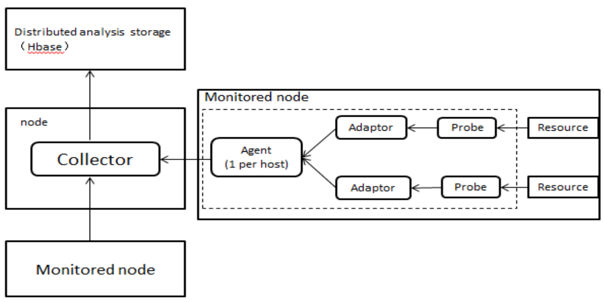

FIG 2 THE STRUCTURE OF THE DISTRIBUTED CLUSTER MONITORING SYSTEM

Distributed cluster data monitoring system collects basic platform information through Chukwa in real-time, including CPU, memory, disk, network

bandwidth, service availability and response time, etc. Then the data are stored in the distributed database. The structure of the distributed cluster monitoring system is shown in Fig 2

Each monitoring node in the cluster consists of a collection agent service responsible for the real-time data collected and sent the data to a receiver. The collector is responsible for storing these small data blocks into hbase in the form of (key, value), where the key is a unique identification number to each data block and the value is a tuple of resource use efficiency (timestamp, hostname, service, process, performance index, the actual value, etc).

Collect Agent: A Collect Agent consists of an adapter and an agent. An adapter is a resource data collection unit and event analysis, a testing body running on the host independently. The agent located on every node that needs monitoring in a Hadoop cluster, responsible for collecting all data on the node, completing the production data, troubleshooting, tasks integrating, etc.

Collector: Transceiver has divided the Hadoop cluster monitoring network into multiple regions, each of which consists of a set of transceiver and multiple data collection agency. Its main task is to receive the data agent from the data collector, sorte and pretreat and then transfer them to the HDFS. Because if only one transceiver corresponds to a region, there may be a single point failure problem. So transceiver take redundancy strategy, that is, an area can have multiple transceivers. The data collection agent will randomly select one transceiver to transmit data when it is collecting data. This can effectively achieve load balancing.

Distributed Runtime Monitoring System

The distributed runtime monitoring system is responsible for the acquisition of data from HBase and transfer of these data into the target item in SLA agreement that has the same meaning. With these data, the distributed runtime monitoring system can send a warning to the QoS knowledge base whenever a SLA violates risk takes place. The structure of the distributed runtime monitoring system is shown in Fig 3.

The distributed runtime monitoring system employs a series of Pig scripts running on different nodes to perform analysis tasks. Pig has rich data structures to process large data sets. Each pig script is compiled into a series of map-reduce tasks to parallel processing input data and write data. The vast resources data collected from the platform is transmitted to the runtime monitoring system and the QoS knowledge base. Once the related data is collected, the runtime monitoring system will transform these data into the related parameters in the SLA agreement.

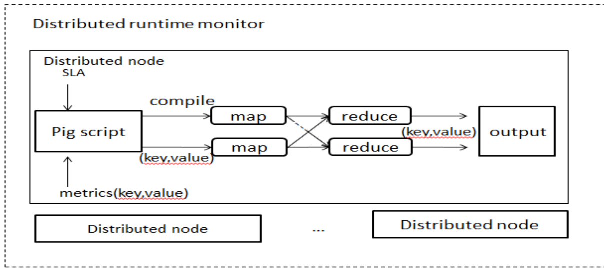

FIG 3 THE STRUCTURE OF THE DISTRIBUTED RUNTIME MONITORING SYSTEM

Service availability in the SLA parameters list, for example, can be calculated by means of downtime and online time. When in SLA defaults, run-time monitoring will inform Qos knowledge base, knowledge base components for predefined risk threshold and use the resource scheduler to make corresponding adjustments.

QoS Knowledge Base

This article uses the measurement method of lower resource metrics to senior SLA designed in the literature[6]. This method mapped the results of the resource metrics into predefined SLA parameters list and utilized Case-Based Reasoning (CBR) to make decisions. A complete case is shown in Fig 4.

```
(
  (Job,1)
  (
    (
      (Incoming Bandwidth , 12.0) .
      (Outgoing Bandwidth , 20.0) .
      (Storage , 12000) .
      (CPU Time , 500) .
      (Memory , 2000)
    ),
    "Increase Memory 200MB"
    (
      (Incoming Bandwidth , 12.0) .
      (Outgoing Bandwidth , 20.0) .
      (Storage , 12000) .
      (CPU Time , 500) .
      (Memory , 2200)
    ),
    0.002
  )
)
```

FIG 4 A COMPLETE CASE OF QOS KNOWLEDGE BASE

Resource Scheduler

It's very important for Hadoop to dynamically schedule the resource of the basic platform. The

resource scheduler based on SLA agreement uses Hadoop's own scheduler to schedule the resource according to the orders from QoS knowledge base. This can effectively avoid the occurrance of the SLA violation threats.

Experiments and Performance

Experiment Environment

In order to validate the proposed architecture, this paper has built a prototype system based on the system structure this article proposed. This system is based on a Hadoop cluster with eight hosts; each of which has an i5 CPU, a memory of 2 GB, a disk of 500 GB. The cluster has seven slave nodes and one master node; each of which is running on Ubuntu10.04 with a JDK 1.6.0 _21; Hadoop version: Hadoop - 1.0.0; and Hadoop configuration file is 1 GB Heap space. Experimental hypothesis is that different users submit assignments and calculation and store the data into Hadoop.

Real-time Performance Evaluation

In order to check the real-time performance of the system, according to different host number, the system monitors and deals with each cloud resource index databy recording system processing time of each module. The T_map is the underlying index map to the top of the SLA parameters; and T_process is default detection processing time, while T_res is warning scheduling time. System overall running time T_total is needed for the call = T_map + T_res +T_process.

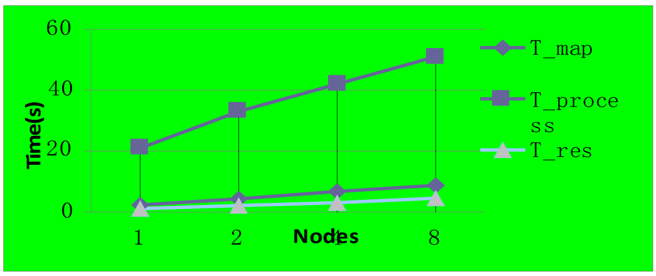

FIG 5 TIME SPENT WITH DIFFERENT NODES

From Fig 5 it can be observed that T_process and T_map take the most time. Because the cloud has mass data and SLA agreement, Pig is employed to parallelling process these jobs. As the number of the machines increases, the system can process the jobs more efficiently. T_res warning time is shorter, usually less than 30 s. T_map time controls over the small scope, and the result shows that for massive amounts of data monitoring, the system has good real-time performance and scalability.

Service quality Evaluation

In order to validate that the proposed system can be very good for the possibility to avoid SLA breach. The proposed system is used to collect and analyze the data in a cloud environment. Then the QoS knowledge base is in use to make decision and the possible order is sent to the Resource scheduler to dynamically rearrange resources. We use the most common scenario in Hadoop, upload files, as the test experiment. Whenever the folder size reaches the SLA specified range, the upload task will fail if the proposed system doesn't move these files in advance.

The initial conditions of the experiment are as followed: 2 users, each uploads a file ranging from 2 to 5 MB every 30 to 60 seconds. The System checked the folder size every 20 seconds. The folder size ceiling is set to be 330 MB (64 MB*5+10 MB). The threshold of the system is set to be 260 MB (330 MB-70 MB).The experiment has run for 56 hours and the result is as followed:

TABLE 2 SERVICE PERFORMANCE EVALUATION

Indicators	Without SLA	With SLA
The success rate	98.71%	100%
Success/Total counts	4140/41941	4168/4168
Successfully saved data	15207497728 Byte	15267266560 Byte
The total upload data	15454961664 Byte	15268315136 Byte
Data integrity	98.40%	99.99%

As it can be seen from the table that with SLA, operation success rate is nearly 100%, and without SLA, no matter how perfect a job is, it always will fail from time to time. The SLA can obviously improve the integrity of the data. A loss of data only happens when the SLA time and operation time collide, which can be weaken by reducing SLA interval detection time.

Conclusions

In this article, a system collecting the data from a cloud computing platform with massive resources having been put forward was monitored in real-time. The system also provided a distributed runtime monitoring service that can perform a quick mapping monitoring based on massive SLA agreement and dynamically rearrange resources. The experimental results verified the feasibility and effectiveness of the proposed system. Next, focus will be placed on machine learning based on cloud computing under vast amounts of knowledge base to ensure the QoS more intelligent, as the scheduling system is more purposeful.

REFERENCES

Boulon J, Konwinski A, Qi R, et al. Chukwa, a large-scale monitoring system[C].Proceedings of CCA. 2008, 8.

Comuzzi M, Kotsokalis C, Spanoudakis G, et al. Establishing and monitoring SLAs in complex service based systems[C].Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009: 783-790.

Frutos H M, Kotsiopoulos I. BREIN: Business objective driven reliable and intelligent grids for real business [J]. International Journal of Interoperability in Business Information Systems, 2009, 3(1): 39-42.

Gunter D, Tierney B, Crowley B, et al. Netlogger: A toolkit for distributed system performance analysis[C].Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proceedings. 8th International Symposium on. IEEE, 2000: 267-273.

Khetrapal A, Ganesh V. HBase and Hypertable for large scale distributed storage systems [J]. Dept. of Computer Science, Purdue University, 2006.

Maurer M, Brandic I, Emeakaroha V C, et al. Towards knowledgemanagement in self-adaptable clouds[C].Services (SERVICES-1), 2010 6th World Congress on. IEEE, 2010: 527-534.

Patel P, Ranabahu A, Sheth A. Service level agreement in cloud computing[C].Cloud Workshops at OOPSLA. 2009.

Theilmann W, Yahyapour R, Butler J. Multi-level sla management for service-oriented infrastructures [M].Towards a Service-Based Internet. Springer Berlin Heidelberg, 2008: 324-335.